



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) EP 0 707 263 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
17.04.1996 Bulletin 1996/16

(51) Int Cl.⁶: G06F 9/44

(21) Application number: 95306535.6

(22) Date of filing: 15.09.1995

(84) Designated Contracting States:
DE FR GB

(30) Priority: 19.09.1994 US 308190

(71) Applicant: International Business Machines
Corporation
Armonk, N.Y. 10504 (US)

- Calvert, Nathaniel
Rochester, Minnesota 55901 (US)
- McBride, Thomas Marcus
Rochester, Minnesota 55901 (US)
- Snyder III, Arthur Frank
Rochester, Minnesota 55901 (US)
- Ure, John Robert
Coral Springs, Florida 33067 (US)

(72) Inventors:
• Bramnick, Arnold H.
Boca Raton, Florida 33433 (US)

(74) Representative: Moss, Robert Douglas
IBM United Kingdom Limited
Intellectual Property Department
Hursley Park
Winchester Hampshire SO21 2JN (GB)

(54) PC software package registration and tracking

(57) A method for application registration and tracking for DOS and OS/2 software independent of the mechanism in which the application is distributed and installed. The invention allows other application registration and tracking techniques to co-exist without affecting/impacting each other. This truly is an "open" packaging, registration and tracking invention which allows PC-software to be better managed in a network.

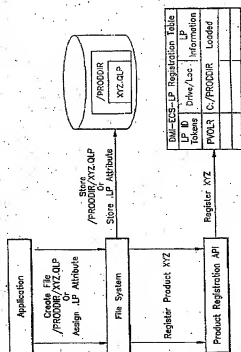


FIG. 2

EP 0 707 263 A1

Description

Field of the Invention

This invention relates to a method of managing computer software. More specifically, this invention relates to registering and tracking replaceable software units configured in a multi-level hierarchical system.

The present invention uses the basic packaging structure of U.S. patent 5,237,688, entitled SOFTWARE PACKAGING STRUCTURE HAVING HIERARCHICAL REPLACEABLE UNITS, commonly assigned and hereby incorporated by reference.

The 5,237,688 patent is directed to a method for identifying how an application may be converted to a "packaged application" via a set of machine functions. The present invention uses the benefits of this packaging technique to provide a new way to track software units.

BACKGROUND OF THE INVENTION

Today software on a PC may be installed from many sources via electronic media (e.g. online, databases, networks, services, bulletin boards, etc.) or magnetic/optical media (e.g. floppy disks, CD-Rom, etc.). It is desirable to register and track software when it is installed to insure that software registration licenses are not violated. Owners and/or users of distributed applications would like to know the exact maintenance level of distributed software so that a central site can determine if the latest version of the software exists on a particular PC. The PC must have some mechanism of registering and tracking software which is resident on the PC. To accomplish this, the application registration and tracking facility must track the movement of the applications' files within the PC's directories and sub-directories. The prior art has failed to provide for an automatic method for tracking distributed and/or hierarchical software units.

The software units of the present invention may be software application packages made up of several linked replaceable units (RU). Each RU is serviceable without adversely affecting the other RUs. RUs are linked together in a hierarchical fashion in a series of levels. In the preferred embodiment, five levels are used: Application Group level (AG), Loadable Code Group level (LCG), Primary Functional Group level (PFG), Secondary Functional Group level (SFG) and Operational Code Group level (OCG). The AG level defines a group of computer programs combined to perform a high level application tailor fit to meet the needs of the user. The LCG level defines individual programs each created to perform a general task. The PFG level refines the common programs defined in the LCG level to a more specific set of primary functions. The SFG level refines the primary functions defined in the PFG level to an even more specialized set of secondary functions tailored closely to fit a specific user's needs. The OCG level contains the operational code needed to run the specialized user appli-

cation package defined by the preceding four levels. A more complete discussion of packaging techniques is provided for in the 5,237,688 patent.

The present application may reference terms that are different than the terms specifically defined in patent 5,237,688. The following is a mapping of terms defined in this application to those defined in the referenced patent.

Application Group (AG) is equivalent to a Suite of Licensed Programs (LPs)

Loadable Code Group (LCG) is equivalent to an ECS-LP or a product

Primary Functional Group (PFG) is equivalent to all ECS-LP Loads with the same functional group ID (Feature ID).

Secondary Functional Group (SFG) is equivalent to a specific optional load which may contain both translated software (NLV) and non-translated software. The SFG is therefore equivalent to the ECS-LP-Load which may be translated or non-translated. The Operational Code Group (OCG) is equivalent to functional files shipped with a translated or non-translated ECS-LP-Load. Figure 1 illustrates how packaging in the present application maps to that defined in the referenced patent (5,237,688).

SUMMARY OF THE INVENTION

The present invention provides a method of automatically tracking computer software units in a computer file system. Including an operating system and operating system file directories, comprising: uniquely identifying said computer software units; creating a tracking directory recognizing said uniquely identified computer software units; monitoring said computer file system for changes to said operating system file directories; detecting changes in said operating system file directories involving said uniquely identified computer software units; and updating said tracking directory to reflect said changes in said uniquely identified computer software units.

The present invention facilitates automatic management of software, and in particular facilitates tracking of the installation and distribution of replaceable, packaged software units in a hierarchical file system. The invention may be used to manage software in a client/server, network and/or distributed systems environment. The invention may be used to automatically register and track software as it is loaded and/or moved throughout a hierarchical file structured system.

The present invention preferably defines a mechanism for registering and tracking packaged applications on a hierarchical file system environment (i.e. DOS, OS/2), but is not limited thereto. The present invention preferably provides for a specific operating system implementation for an application registration and tracking API (ARTA) which allows a product (i.e. ECS-LP) tracking system to "know" about the software installed on the

PC independent of the directory or sub-directory in which the application exists. The invention preferably also provides flexibility in allowing an application to be installed into a customer targeted directory without compromising its ability to be registered and/or tracked.

This present invention is dependent on the ability to package an application with control information containing the software packaging structures of the 5,297,688 patent. In a preferred embodiment, the invention further allows a PC application to be packaged without recompilation of the existing PC application. The packaging control structure is assigned a distinguishing file suffix.

BRIEF DESCRIPTION OF THE DRAWINGS(S)

The present invention will now be described in more detail, by way of example, with reference to the accompanying drawings in which:

Figure 1 shows ECS-LP Packaging Structures. An Electronic Customer Support (ECS) enabled Licensed Program (LP) is packaged according to this figure.

Figure 2 illustrates a general flowchart of a registration technique according to the present invention.

Figure 3 illustrates a packaging technique of an Electronic Customer Support(ECS) enabled Licensed Program(LP).

Figure 4 illustrates Electronic Customer Support (ECS)-Licensed Program(LP) packaging formats.

Figure 5 illustrates Electronic Customer Support (ECS)-Licensed Program(LP) fix packaging formats.

DETAILED DESCRIPTION

The present invention manages and tracks packaged software units implemented in a typical DOS or OS/2 environment, but should not be limited thereto. It is envisioned that the present techniques could be modified to be useful in any operating system.

A DOS or OS/2 operating system provides an interface to intercept all modifications to directories. Modifications include create, rename, delete or a combination of either. A DOS or OS/2 operating system further provides an interface to intercept all file create, rename, delete and move operations.

For DOS the intercept will be done via an interrupt handler, supplied by ARTA, which monitors for the aforementioned instructions. For OS/2 the intercept will be done by the OS which monitors for the API and determines if the ARTA intercept handler has been activated in the system.

ARTA will build a product directory for all locally attached directories which contain a PRD file suffix. The PRD suffix is an arbitrary value determined and set by a packaging suffix manager. It could be any value. The file containing the suffix name is flagged as containing LP description information or LP Fix description information. Note, ARTA ensures that files which have a suffix name of PRD contain the correct format prior to updating the

ARTA product directory.

The following functions are performed, by the ARTA intercept handler according to the operation being performed:

a. Copy of a Directory which contains a file name which contains a PRD suffix.

b. Rename of a directory which contains a file name in which contains the PRD Suffix

Update the ARTA product directory to reflect the new directory which contains the PRD File Suffix

c. Creation of a file which has a suffix name PRD

Validate that the file with the PRD File Suffix contains packaging information. If it does, update the ARTA product directory.

d. Rename of a file which contains a PRD File Suffix name

Update the prefix file name in the ARTA product directory. If the PRD File Suffix is renamed, remove the product from the ARTA product directory.

e. Delete of a file which contains a PRD File Suffix name.

Remove the file from the ARTA product directory.

ARTA also provides a retrieve product data API which allows another operating system program to retrieve product information from the ARTA product directory.

DESCRIPTION OF THE ILLUSTRATIVE EMBODIMENT(S)

ECS-LP Functional Packaging and Registration

The preferred embodiment of the present invention uses the packaging techniques as disclosed in the 5,297,688 patent.

Electronic Customer Support(ECS) enabled Licensed Program (LP) (ECS-LP) Functional Package

Electronic Customer Support(ECS) enabled Licensed Program(LP) (ECS-LP) Functional Package supports Primary and Secondary Features/Options. The Primary Feature contains an LP instance Description. All Options of any Feature will contain one or more LP Load instance Description files. There will be one for non-translatable load instance (code) and one for each translatable load instance (NLV). The ECS-LP packaging may be done without recompiling functional files of the application/ECS LP.

ECS-LP Registration Facility

The ECS-LP Registration Facility allows multiple instances of an ECS-LP to exist in Transport state and further allows multiple instances of an ECS-LP to exist in a Loaded state. Under certain circumstances, the ECS-LP Registration Facility allows multiple instances of an ECS-LP to exist in the Installed state. Multiple instances will be allowed in the Installed state if:

1. ECS-LP is enabled for ECS-LP replication (Multiple versions of ECS-LP with the same maintenance level installed at the same time) or
2. ECS-LP is enabled for multiple versions of the ECS-LP installed on the system with different maintenance levels.

ECS-LP Registration API is activated each time the path to a LP instance is changed (renamed/moved/deleted). The Registration Facility works with local drives, network drives, and removable media or other equivalent means.

ECS-LP Release Package Registration Flow

The following is a typical flow for the Creation of an ECS-LP Release Package:

Use desk top tool kit to assemble ECS-LP functional files;

Use ECS-LP packaging APIs to create ECS-LP Instance Description file (Creation registers the description);

Use ECS-LP packaging APIs to create ECS-LP Load Instance Description File(s);

One ECS-LP Load Operation File for non-translatable software (code);

One ECS-LP Load Description File for translatable software (NLV) if the software is language sensitive;

The Creation function registers the ECS-LP description file(s) as shown in figure 2, wherein:

the PRD Extension of ECS-LP File attribute triggers File System;

the File System calls ECS-LP Registration API;

the ECS-LP Registration API updates a ECS-LP Registration Table.

All files are packaged into a distribution transport directory;

Add installation facility to ECS-LP in the Transport format to the Install format;

Add ECS-LP Instance Description file and ECS-LP Load Instance File(s) to a distribution transport directory, see Figure 4. This registers the ECS-LP description file(s) with the ECS-LP transport format.

Distribute ECS-LP in Transport format include any of the following: CD-ROM image, Diskette image, Electronically or any equivalent means;

Install ECS-LP from Transport format by:

using an installation facility which extracts and installs functional files from the transport format. The installation facility also installs the ECS-LP Description files;

zipped ECS-LP Instance Description files which are also unzipped and installed from ECS-LP Transport format by desk top Common Installer.

This above sequences register the ECS-LP description file(s) installed with ECS-LP. See Figures 2 and 4.

LP Fix Packaging

A possible format of a ECS-LP fix package is defined in Figure 5. The possible format contents of a ECS-LP Fix package required for registration is defined therein. The ECS-LP fix package may be applied to any ECS-LP Load as shown in Figure 3 and outlined below:

Fix to a ECS-LP Load will contain on LP Fix Description;

A fix can be for a replaceable unit (file) in a:

Non-translatable ECS-LP Load (Code)

Translatable ECS-LP Load (NLV)

ECS-LP Fix Registration Facility

The ECS-LP Fix Registration Facility allows multiple instances of an ECS-LP Fix to exist in Transport state. The ECS-LP Fix Registration Facility further allows multiple instances of an ECS-LP Fix to exist in Loaded state. In certain circumstances, multiple instances of an ECS-LP Fix exist in the Installed state if:

1. ECS-LP is enabled for ECS-LP replication (Multiple versions of ECS-LP with the same maintenance level installed at the same time), or

2. ECS-LP is enabled for multiple versions of the ECS-LP installed on the system with different maintenance levels.

ECS-LP Registration API is activated each time the path to a LP Fix Description is changed (e.g. renamed/moved/deleted). Please note, the Registration Facility works with local drives, network drives, removable media or other equivalent means.

Create ECS-LP Fix Package

The following descriptions provide an outline for the creation of an ECS-LP Fix Package:

Use desk top tool kit to assemble ECS-LP functional files to be fixed;

Use ECS-LP packaging APIs to create ECS-LP Fix Instance Description file (Creation registers the description);

Creation registers the ECS-LP Fix description file (s);

PRD Extension or ECS-LP Fix File attribute triggers File System

File System calls ECS-LP Registration API
ECS-LP Registration API updates ECS-LP Registration Table, (See Figure 2)

Package all Fix files into a distribution transport directory;

Add installation facility to ECS-LP fix Transport format;

The installer will transpose the ECS-LP Fix in the Transport format to the Install format;

Add ECS-LP Fix Instance Description file to the distribution transport directory (see Figure 5)

The above sequence registers the ECS-LP Fix description file with ECS-LP Fix Transport format.

Distribute ECS-LP Fix in Transport format includes any of the following-CD-ROM image, Diskette image, Electronically or any equivalent means;

Install ECS-LP FIX from Transport format:

An install facility extracts and installs ECS-LP Fix functional files from Transport format

The ECS-LP Fix Instance Description file is installed from ECS-LP Fix Transport format by an install facility

This registers the ECS-LP Fix description file installed with ECS-LP Fix. (See Figures 2 and 5).

Retrieve ECS-LP Information API

Retrieval of ECS-LP Information API provides data in three formats:

1. Compressed list format
2. Expanded Registration Table format
3. Expanded Description file format

Locates ECS-LP Information based on parameters passed.

Retrieve ECS-LP PTF Information API

Retrieval of ECS-LP PTF Information API Provides data in three formats:

1. Compressed list format
2. Expanded Registration Table format
3. Expanded Description file format

Locates ECS-LP PTF information based on parameters passed.

CONCLUSION

Since ECS-LP uses packaging files, these files may

coexist with other packaging and registration protocols. ECS-LP registration API also provides a capability for registering ECS-LPs and ECS-LP Fixes into other desk top management facilities without forcing the application developer to write or distribute registration APIs with the application. The advantage of this approach is the application does not have to be redistributed each time a new registration management facility is installed or updated. Another advantage of this approach is ECS-LPs and their fixes can be registered in the transport format when it is not possible to activate application APIs.

A system and method has been shown in the above embodiments for a unique method to register and track software units. While various preferred embodiments have been shown and described, it will be understood that there is no intent to limit the invention by such disclosure, but rather, is intended to cover all modifications and alternate constructions falling within the spirit and scope of the invention as defined in the appended claims.

Claims

1. A method of automatically tracking computer software units in a computer file system, including an operating system and operating system file directories, comprising:
uniquely identifying said computer software units;
creating a tracking directory recognizing said uniquely identified computer software units;
monitoring said computer file system for changes to said operating system file directories;
detecting changes in said operating system file directories involving said uniquely identified computer software units; and
updating said tracking directory to reflect said changes in said uniquely identified computer software units.
2. A method according to claim 1, wherein said computer software units comprise replaceable packaged application programs.
3. A method according to claim 2, wherein said packaged applications are tracked independent of file attribute capabilities of said operating system environment.
4. A method according to any one of claims 1 to 3, wherein said uniquely identified software units are identified by a name and/or attribute.
5. A method according to claim 4, wherein said unique identification of said computer software units further comprises appending a distinct file name suffix.

6. A method according to any one of the preceding claims, wherein said changes in said file directories include renaming, moving and/or deletion of directories.
7. A method according to any one of the preceding claims wherein said changes in said file directories include renaming, moving and/or deletion of files.
8. A method according to any one of the preceding claims, wherein said tracking directory is importable/exportable to additional systems.
9. A method according to any one of the preceding claims, which operates on local disk drives, network drives and/or removable media.
10. A method according to any one of the preceding claims, wherein the computer software units comprise non-translatable software.
11. A method according to any one of claims 1 to 9, wherein the computer software units comprise translatable software.

5

10

15

20

25

30

35

40

45

50

55

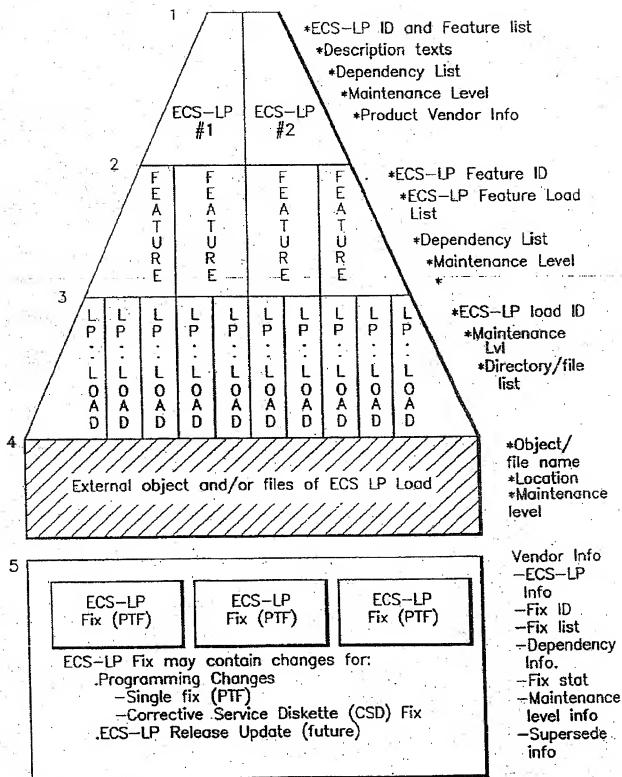


FIG. 1

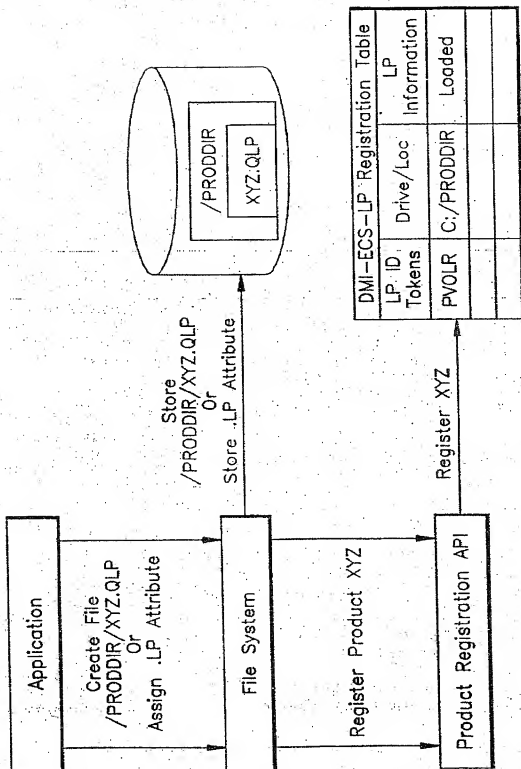


FIG. 2

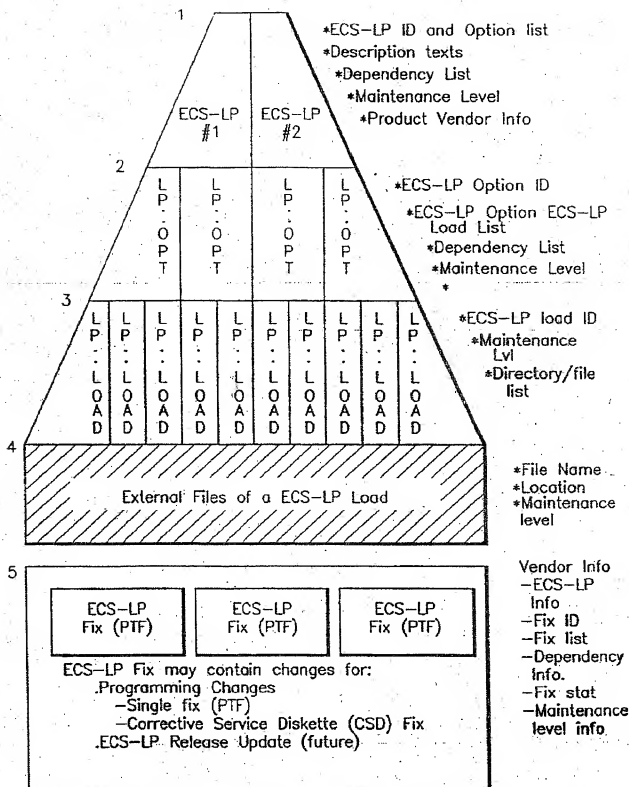


FIG. 3

FIG. 4

LP Transport Format:

Application ZIP Files Contain: o Application Files o LP Description Files	Install Program	LP Description File o DMI-ECS-LP Instance Description File o DMI-ECS-LP Load Instance Description File
---	------------------------	---

LP Load/Install Format:

Drive/Directory/SubDir o DMI-ECS-LP Instance Description File o DMI-ECS-LP Load Instance Description File o LP Application Files
--

FIG. 5

LP Transport Format:

Application Fix ZIP Files Contain: ◦ Application Fixes ◦ LP-Fix Description File	Install Program	LP-Fix Description File ◦ DMI-ECS-LP-Fix Instance Description
---	-----------------	--

LP Load/Install Format:

Drive/Directory/SubDir ◦ DMI-ECS-LP-Fix Instance Description File ◦ LP Application Fixes
--



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 95 30 6535

DOCUMENTS CONSIDERED TO BE RELEVANT		
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim
A	US-A-5 023 907 (JOHNSON HERRICK J ET AL) 11 June 1991 * column 2, line 12 - line 16 * * column 2, line 49 - column 3, line 10 * * column 3, line 34 - line 55 * * column 4, line 14 - line 25 *	1-11
A	US-A-4 558 413 (SCHMIDT ERIC E ET AL) December 1985 * column 1, line 12 - line 36 * * column 2, line 25 - line 52 * * column 9, line 28 - line 59 *	10 1-11
<p>TECHNICAL FIELDS SEARCHED (Int.Cl.4)</p> <p>G06F</p>		
The present search report has been drawn up for all claims		
Place of search	Date of completion of the search	Examiner
THE HAGUE	18 December 1995	Brandt, J
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X: particularly relevant if taken alone Y: particularly relevant if combined with another document of the same category A: technological background O: non-written disclosure P: intermediate document</p> <p>T: theory or principle underlying the invention E: earlier patent document, but published on, or after the filing date D: document cited in the application L: document cited for other reasons</p> <p>d: number of the same patent family, corresponding document</p>		

EPO FORM 183 (04.92) (PUBLISHED)